Decremental Learning of Evolving Fuzzy Inference Systems Using a Sliding Window

Manuel Bouillon, Eric Anquetil, Abdullah Almaksour

Decremental Learning of Evolving Fuzzy Inference Systems Using a Sliding Window

Manuel BOUILLON, Eric ANQUETIL, Abdullah ALMAKSOUR
INSA de Rennes, Avenue des Buttes de Coesmes, F-35043 Rennes
CNRS, UMR IRISA, Campus de Beaulieu, F-35042 Rennes
Université Européenne de Bretagne, France
{*manuel.bouillon, eric.anquetil, abdullah.almaksour*}*@irisa.fr*

***Abstract*—This paper tackles the problem of decremental learning of an evolving classification system. We study the use of decremental learning to improve performance of evolving recognizers in non-stationary scenarios. Our on-line recognizer is based on an evolving fuzzy inference system. In this paper, we propose a new strategy to introduce decremental learning, with the use of a sliding window, in the optimization of fuzzy rules conclusions. This approach is based on a downdating technique of least squares solutions for unlearning old data. This technique is evaluated on handwritten gesture recognition tasks. In particular, it is shown that this downdating techniques allow to adapt to concept drifts and that we face a precision reactiveness trade-off. It is also demonstrated that decremental learning is necessary to maintain the system learning capacity over time, making decremental learning essential for the life-time use of an evolving classification system.**

***Keywords*-Incremental Learning; Decremental Learning; Evolving Fuzzy Inference System; Recursive Least Squares; Concept Drifts; On-line Gesture Recognition**

I. INTRODUCTION

Evolving classification systems have appeared in the last decade to meet the need for recognizers that work in changing environments. They use incremental learning to adapt to the data flow and to cope with class adding (or removal) at runtime. This paper focuses on the decremental learning of such an evolving recognizer, based on a fuzzy inference system. The aim of decremental learning is twofold. First, to maintain the system learning capacity, and second, to forget obsolete data and focus the learning process on current environment.

The target application of this work is the use of on-line handwriting gesture classifiers to facilitate interactions with computers using pen-based interfaces like tablet computers, smartphones, whiteboards, etc. Gestures can be drawn differently from one user to another, and users may want to add or remove gestures, as long as they use the application. Moreover, users would often change progressively the manner by which they draw gestures. Novice users start drawing carefully and slowly their gestures, while they do them in a more fluid and rapid manner as they become expert. To cope with these requirements, forgetting capacity must be used to increase system reactivity and performance in such dynamic environment.

We extend in this paper our evolving classification system *Evolve* [1] by integrating a decremental learning technique. A new decremental strategy, relying on a sliding window of data samples, is proposed. This window is used to unlearn old data – so-called downdating of least squares solutions.

We present briefly the architecture of *Evolve* and its incremental learning algorithm in Section II. Our proposed decremental learning method is detailed in Section III. This new approach is then evaluated on some handwritten gesture recognition tasks in section IV. Section V concludes and discusses future work.

II. SYSTEM ARCHITECTURE

We focus here on Fuzzy Inference Systems (FIS) [2], which have demonstrated their good performance for classification for quite some time. Moreover, they can easily be trained incrementally (in real time) and have a good behavior when new classes are added. [3] and [1] are recent examples of evolving FIS used for on-line classification.

Our evolving system *Evolve* [1] is a first-order Takagi-Sugeno (TS) fuzzy inference system. It consists of a set of fuzzy inference rules. Rule premises are the fuzzy membership to the rule prototype, which are clusters in the input space. Rule conclusions are fuzzy membership to all classes, as a linear function of the input, that are combined to produce the system output.

In *Evolve*, both the rule premises and conclusions are incrementally adapted. Rule prototypes are statistically updated to model the runtime data. Rule conclusions parameters are optimized on the data flow, using Recursive Least Squares (RLS) algorithm [4]. New rules, with their associated prototypes and conclusions, are created by an incremental clustering method when needed.

As rule conclusions induce decision boundaries, we decided to introduce decremental learning in the conclusions optimization process, namely the RLS algorithm.

III. DECREMENTAL LEARNING OF EVOLVING FUZZY INFERENCE SYSTEMS

This paper focuses on decremental learning for the life-long use and learning of an evolving classification system

optimized with the Recursive Least Squares (RLS) algorithm.

The interest of integrating forgetting in a learning system is twofold. First, decremental learning is necessary to limit the weight of past data, and thus to maintain the learning capacity of the system. If every data sample has the same relative weight, the system will tend to become set with time. Second, decremental learning allows the system to follow any change – concept drift [5] – of the data flow by unlearning obsolete data samples.

Several decremental systems already exist in different contexts. The use of a forgetting factor in RLS algorithm [6] is well known in the literature dedicated to control of complex systems. It comes to using the RLS algorithm on an (exponentially weighted) sliding window. However, this algorithm behaves poorly when systems are not uniformly exited – it is known as the covariance "wind-up" problem [7] – which is the case in classification problems.

We present here a new decremental learning strategy applied to FIS optimization with the RLS algorithm. We propose a new decremental learning approach based on downdating the least squares solutions with a sliding window. We use a real time algorithm to unlearn old data that leave the window: the de-recursive least squares algorithm.

A. Principle

The principle of this approach is simple, we maintain a sliding window over the latest data, and we optimize the rules conclusions only on this window of data.

As we can't afford to rebuild rules conclusions at the arrival of each new data, they are updated using the recursive least squares (RLS) algorithm [1]. In the same way, we need to downdate the least squares solutions at departure of each old data from the window, without complete rebuilding. To do so, we propose to use the de-recursive least squares algorithm in order to recursively unlearn old data.

B. De-Recursive Least Squares (DRLS) Algorithm

Let $\theta^{(i)}_{s\to t} \in \mathbb{R}^{n\times c}$ be the i^{th} rule conclusion, optimized on data samples from (x_s, y_s) to (x_t, y_t) (with $x_k \in \mathbb{R}^n$ the input vectors, and $y_k \in \mathbb{R}^c$ the binary output vectors). We propose to downdate least squares solutions with the following DRLS formulas to unlearn "old" data sample (x_s, y_s).

$$\theta^{(i)}_{(s+1)\to t} = \theta^{(i)}_{s\to t} - P^{(i)}_{(s+1)\to t} x_s \beta^{(i)}_s (y_s - x_s^T \theta^{(i)}_{s\to t}) \quad (1)$$

Where the covariance matrices $P^{(i)}$ are updated as follows.

$$P^{(i)}_{(s+1)\to t} = P^{(i)}_{s\to t} + \frac{P^{(i)}_{s\to t} x_s x_s^T P^{(i)}_{s\to t}}{\frac{-1}{\beta^{(i)}_s} + x_s^T P^{(i)}_{s\to t} x_s} \quad (2)$$

These DRLS formulas can easily be proven in a similar way to the standard RLS formulas.

C. Discussion of the Window Size

The sensitive issue of this approach is the size of the sliding window. Indeed, performance is directly linked to the window length. In steady environment, the longer is the window, the lower is the error rate (until a minimum rate). However, a too long window reduces system reactiveness and thus deteriorates performance in changing environment.

We focus on having a large enough window to avoid performance reduction in stationary environment. Such a large window is generally sufficient to limit old data weight and to adapt to concept drifts in non-stationary environment. Even if a fixed length sliding window is not optimal, it provides a good behavior as it will be demonstrated experimentally.

The window minimum length is a problem dependent variable that must be empirically determined. We recommend to dynamically link the window size to the number of classes (noted c) to avoid performance scale-down when new classes are added to the system.

IV. Experimental Results

Starting from the state-of-the-art recognizer *Evolve* [1], we implemented our new approach: *Evolve D* – with Downdating. All systems use the Heterogeneous Baseline Feature set (HBF49) [8].

In this section, we first present the incremental evaluation protocol, and the datasets, used for testing. Then, we measure system inertia to novelty to asses the need for decremental learning. Next, we study the impact of the window length in both stationary and non-stationary environment.

A. Incremental Evaluation Protocol

To evaluate our systems in a realistic way, we used an incremental evaluation protocol called predictive sequential – or prequential [9] – with a sliding window to converge to the holdout error.

As an incremental system first tries to recognize a data sample, and then learn from it once it has the true label, we evaluated our systems in a similar way. Each data sample is first used as test sample, and then as learning sample. Error rates are then computed between every test points.

B. Datasets

As this work is applied to on-line handwritten gesture recognition, we evaluated our new approach on handwritten gestures databases: ILGDB[1] [10] and IRONOFF-digit [11].

1) ILGDB: This database contains handwritten gestures that have been collected in an immersive environment. It is composed of 6629 mono-stroke gestures, belonging to 21 classes, which were written by 38 writers. This database is very interesting for several reasons.

First, gestures are ordered chronologically in their drawing order which allows us to see changes in writer style

[1]Freely available at http://www.irisa.fr/intuidoc/ILGDB.html

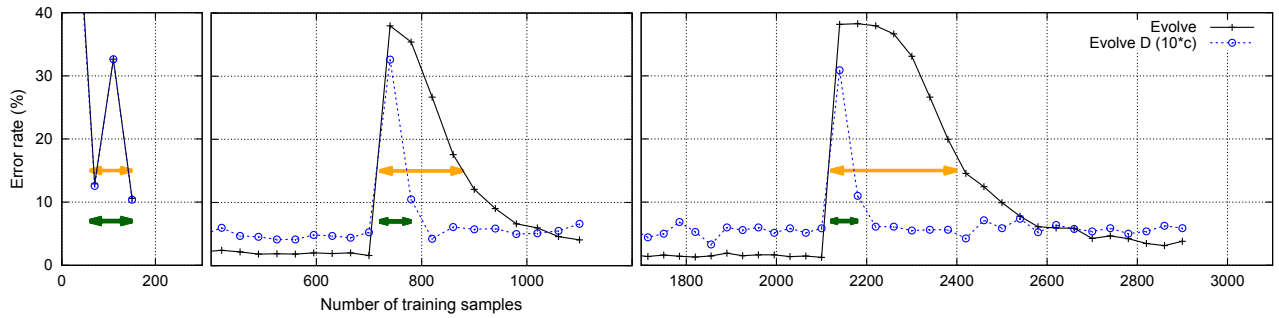

Figure 1: Scenario "inertia and reactiveness" – IRONOFF-digit database – Introducing novelty after 70, 700 and 2100 training samples

with time, as the writer changes from novice to expert. Second, class frequencies varies, from 5 to 17 examples per class. Third, for part of the database, gesture classes are user defined. This features makes this database very realistic and representative of the real use of an on-line recognition system. Some gesture examples are shown in figure 2.

Figure 2: Gesture samples from ILGDB group 1 (free gestures)

The only drawback of this database is the low number of gesture per writer (less than 180) and per class for each writer (from 5 to 17 gestures). To make up for this inconvenience, we also used another database, containing more data, presented below.

2) IRONOFF-digits: The interest of this database is that it offers more gestures (in writer independent mode), but like most classic benchmark databases, IRONOFF-digit is not ordered (no chronological evolution of the data with time).

C. System Inertia to Novelty

We test the reactiveness of the systems, and the evolution of the inertia of their model with time. We train the different systems with seven classes during a varying period time and then introduce three other classes. We measure the time needed by the different systems to learn those new classes. Results averaged over 20 different data orders are shown in figure 1.

If all systems behave similarly after 70 training samples, results are different after 700 or 2100 samples. Looking at the results of our reference system without forgetting *Evolve*, we can clearly see that its reactiveness decreases with time, and that the model tends to become set. After 2100 training samples, the system needs 400 more samples to resume to an error rate under 10%, which makes it partially unusable for quite some time. On the contrary, systems using decremental learning need the same time to learn and adapt their model to the novelty, whenever it is introduced. Their reactiveness is independent of their age.

This test scenario shows the necessity of integrating forgetting into an incremental learning system operating in a non-stationary environment.

D. System Performance in Stationary Environment

We test our different systems on a scenario simulating soft concept drifts. For this purpose, we used the ILGDB 11 writers of group 3 (whose gestures for each class are identical) in a row. We computed the error rate for each writer on the two last phases, approximately the 56 last samples of the 173 of each writer, to measure system performance after the concept drift. Mean results over 20 writer orders are plotted figure in 3.

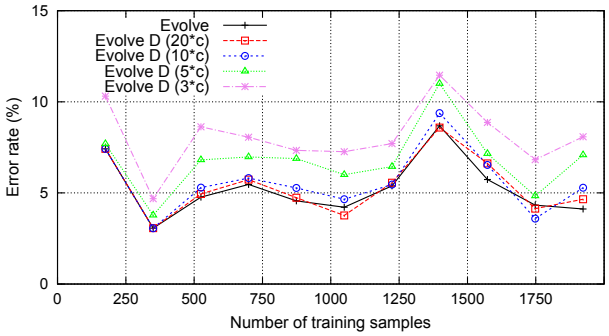

Figure 3: Scenario "soft concept drifts" – Using ILGDB 11 writers of group 3 (fixed gestures) in a row

Writers, and their drawing style, changes but the base gestures of each class stay unchanged. This test scenario is thus nearly stationary and doesn't require the use of decremental learning. Our reference system *Evolve* achieves quite good results here with an average error rate of 5.25%.

Evolve D performance is limited by the restrained number of data it is learning from. With a sliding window of 20 and 10 times the number of classes, *Evolve D* reaches an error rate of 5.38% and 5.61% respectively, which is as good as *Evolve*. A smaller window length reduces performance.

E. System Performance in Non-Stationary Environment

We also test our three systems on a scenario simulating abrupt concept drifts. For this purpose, we used the ILGDB 21 writers of group 1 (whose gestures for each class are different) in a row. We computed the error rate for each writer on the two last phases, approximately the 56 last samples of the 173 of each writer, to measure performance after the concept. Mean results over 20 writer orders are plotted in figure 4.

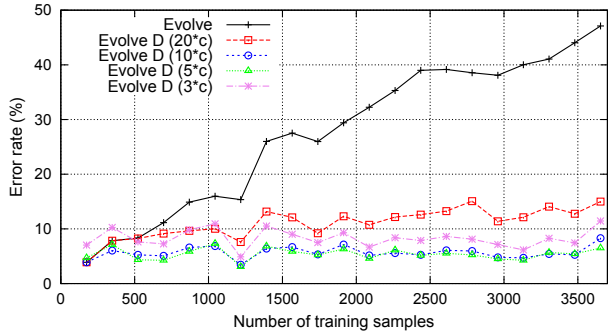

Figure 4: Scenario "abrupt concept drifts" – Using ILGDB 21 writers of group 1 (free gestures) in a row

The results on this scenario are sharply contrasted. Without forgetting, *Evolve* is not able to adapt its model to the changes in the data and ends up with an error rate of 50%.

The use of decremental learning allows *Evolve D* to follow the abrupt concept drifts. We can note that a too long window size (20 times the number of classes) deteriorates performance in this changing environment.

F. Results Discussion

Results on those non-stationary scenarios are quite clear-cut. They show the necessity of using decremental learning to maintain the reactiveness of the system over time. Without downdating, the system model tends to become set, and take ages to learn some novelty. Decremental learning is necessary to maintain system learning capacity.

In the same way, we showed that decremental learning is essential to face (abrupt) concept drifts. Without downdating, the system model becomes more and more complex and performance slowly but surely collapses. Decremental learning allows to discard obsolete data and thus enables the system to focus on current system environment.

We also highlighted the trade-off in the choice of the window length. A short window insures high reactiveness, but deteriorates performance in steady environment. Whereas a long window enables good performance in stationary environment but deteriorates performance in changing environment. For ILGDB, a window length of 10 times the number of classes (210 samples) is a fine compromise that gives good performance in both cases.

V. Conclusion

In this paper, we investigated the decremental learning of an evolving fuzzy inference system used for classification. We proposed a new approach to integrate downdating in the optimization of rules conclusion. Our approach is based on a sliding window and uses de-recursive least squares formulas to decrementally downdate least squares solutions and unlearn "old" data samples from rule conclusions.

This work is applied on handwritten gesture recognition. In particular, we demonstrated that decremental learning is essential to maintain the system reactiveness over time and to adapt to concept drifts. We showed that our approach performs well in changing environment, without deteriorating performance in stationary environment.

Future Work: These approaches could be improved to lessen the accuracy reactiveness trade-off. The sliding window length could be adapted on-line to fit the data flow. The window length could be increased to improve performance during when the data flow is stationary, and reduced when concept drifts are detected.

References

[1] A. Almaksour and E. Anquetil, "Improving premise structure in evolving takagi-sugeno neuro-fuzzy classifiers," *Evolving Systems*, vol. 2, pp. 25–33, 2011.

[2] E. Lughofer, *Evolving fuzzy models: incremental learning, interpretability, and stability issues, applications*. VDM Verlag Dr. Müller, 2008.

[3] P. Angelov and X. Zhou, "Evolving fuzzy-rule-based classifiers from data streams," *Fuzzy Systems, IEEE Transactions on*, vol. 16, no. 6, pp. 1462 –1475, 2008.

[4] S. O. Haykin, *Adaptive Filter Theory (4th Edition)*, 4th ed. Prentice Hall, 2001.

[5] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996.

[6] R. Kulhavy and M. Zarrop, "On a general concept of forgetting," *International Journal of Control*, vol. 58, no. 4, pp. 905–924, 1993.

[7] A. Liavas and P. Regalia, "On the numerical stability and accuracy of the conventional recursive least squares algorithm," *Trans. Signal Processing*, vol. 47, no. 1, pp. 88–96, 1999.

[8] A. Delaye and E. Anquetil, "Hbf49 feature set: A first unified baseline for online symbol recognition," *Pattern Recognition*, vol. 46, no. 1, pp. 117 – 130, 2013.

[9] J. Gama, R. Sebastio, and P. P. Rodrigues, "Issues in evaluation of stream learning algorithms," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 329–338.

[10] N. Renau-Ferrer, P. Li, A. Delaye, and E. Anquetil, "The ILGDB database of realistic pen-based gestural commands," *International Conference on Pattern Recognition (ICPR 2012)*, November 2012, tsukuba (Japan).

[11] C. Viard-Gaudin, P. M. Lallican, P. Binter, and S. Knerr, "The ireste on/off (ironoff) dual handwriting database," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, ser. ICDAR '99, 1999, pp. 455–.